

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Desarrollo de una plataforma
de biología bajo demanda

Autor: Bernardo García Herrera

Director: Alfonso Rodríguez-Patón Aradas

MADRID, JUNIO 2015

ÍNDICE

ÍNDICE	2
1. RESUMEN DEL PROYECTO	3
PROJECT SUMMARY	4
2. INTRODUCCIÓN	5
3. OBJETIVOS	7
4. METODOLOGÍA	10
5. ESTADO DE LA TÉCNICA	12
5.1 Los protocolos biológicos.....	12
5.2 Plataformas Cloud Computing.....	12
5.3 Plataformas de gestión de protocolos biológicos.....	14
5.3.1 Emerald.....	14
5.3.2 Antha.....	15
5.3.3 Arcturus.io.....	15
5.4 Abstracción del modelo presentado por las plataformas.....	16
6. PLATAFORMA DE BIOLOGÍA BAJO DEMANDA	17
6.1 DESCRIPCIÓN.....	17
6.2 ANÁLISIS DEL SISTEMA.....	18
6.2.1 Requisitos del Sistema.....	21
6.2.2 Interfaces comunicación de plataformas.....	28
6.2.3 Caso de Uso de ejemplo.....	29
6.2.4 Arquitectura del sistema.....	30
6.2.5 Patrón de diseño.....	31
6.2.6 Otros aspectos.....	32
6.2 DISEÑO DEL SISTEMA.....	34
6.2.1 Interfaz gráfica para usuarios y administradores.....	34
6.3 IMPLEMENTACIÓN.....	37
6.3.1 Tecnologías.....	37
6.3.1.1 JAVA EE/GLASSFISH.....	37
6.3.1.2 EJB.....	38
6.3.1.3 JPA.....	39
6.3.1.4 REST.....	39
6.3.1.5 JSON.....	40
6.3.1.6 JSP.....	40
6.3.2 Herramientas de desarrollo.....	41
6.3.2.1 Eclipse.....	41
6.3.3 Seguridad.....	42
6.4.3.1 Mecanismos de Seguridad en Java EE/ Glassfish.....	42
6.4 PRUEBAS.....	44
7. CONCLUSIONES	45
8. BIBLIOGRAFÍA	46

1. RESUMEN DEL PROYECTO

La Biología bajo Demanda es un concepto novedoso, que está siendo abordado en la actualidad desde distintos enfoques, que serán expuestos en este documento. Dado este carácter innovador, se trata de un ámbito donde la investigación está muy presente en estos momentos.

Las Tecnologías de la Información y Comunicación (TICs) llevan un tiempo aportando soluciones muy efectivas para algunos de los problemas a los que se enfrenta actualmente la biología sintética. Una de estas soluciones son las plataformas de *Cloud Computing*, que aportan un entorno de trabajo escalable, flexible y seguro. Por ello, se ha empleado este tipo de tecnología en este trabajo fin de grado en el área de la biología sintética mediante el concepto de biología bajo demanda.

Para desarrollar la plataforma de biología bajo demanda ha sido necesario analizar el estado de esta temática actualmente y sus avances. Además, ha sido estimable el estudio de las opiniones de los miembros del grupo de investigación. Todo ello ha permitido llevar a cabo una captura de requisitos adecuada para el ámbito de este proyecto.

Se ha decidido que los servidores de aplicaciones web son la respuesta más adecuada a la hora de implementar las soluciones obtenidas para el desarrollo de la plataforma de biología bajo demanda. En concreto, por sus características, se ha decidido emplear JavaEE de Oracle.

El modelo implementado emplea soluciones conocidas y fiables basadas en patrones de diseño software. Así, conseguimos cumplir con uno de los principales objetivos de este proyecto, que es lograr un sistema flexible y escalable.

Por otro lado, debido a la incertidumbre que conlleva un área tan innovadora, se ha decidido optar por una metodología ágil. Esto supone un plan de trabajo centrado en reuniones semanales conjuntas con el director y los compañeros del grupo de trabajo, empleando prototipado rápido y programación extrema.

Finalmente, se ha conseguido desarrollar una plataforma de biología bajo demanda que puede ser la base para el trabajo de los biólogos del ámbito de la biología sintética en un futuro próximo.

PROJECT SUMMARY

Biology on demand is a new concept, which is currently being addressed from different approaches, which will be presented in this document. Given this innovative character, it is an area where research is a main factor right now.

Technologies of Information and Communication Technologies (ICTs) have provided very effective solutions to some of the problems that synthetic biology is currently facing. One of these solutions is cloud computing platforms, which provide an environment for scalable, flexible and secure work. Therefore, we have used this technology in this final project in the area of synthetic biology through the concept of biology on demand.

To develop a biology-on-demand platform it has been necessary to analyze the state of art. The opinions of members of the research group have also been very influential. All this has allowed us to conduct a proper capture requirements for the scope of this project here developed.

It was decided that web application servers are the best answer when it comes to implementing the solutions obtained for the development of biology-on-demand platform. In particular, by its main features, it was decided to use Oracle's JavaEE.

The implemented model uses known and reliable solutions based on software design patterns. So, we get to meet one of the main objectives of this project, which is to achieve a flexible and scalable system.

On the other hand, due to the uncertainty involved in such an innovative area, it was appropriate to opt for an agile methodology. The work plan was focused on weekly meetings with the director and coworkers, using additive technology and extreme programming.

Finally, this project has been successful in developing a biology-on-demand platform that can be the basis for the work of biologists in the field of synthetic biology in the near future.

2. INTRODUCCIÓN

La Biología Sintética es una disciplina que tiene como principal objetivo el diseño de sistemas biológicos que no existen en la naturaleza. Entre sus principales líneas de aplicación se encuentran el desarrollo de nuevos fármacos, la terapia génica, la reparación y regeneración de tejidos o la reprogramación celular. Potencialmente, su desarrollo supone un cambio de paradigma en la biotecnología actual comparable al impacto que ha tenido la introducción de la informática en las distintas disciplinas científicas.

La biología sintética es una rama inherentemente interdisciplinar en la que deben confluir los conocimientos de varios ámbitos para dar lugar a soluciones eficientes. La ingeniería relacionada con las TICs puede ser una de las ramas que más influya en la biología sintética, aportando su capacidad de facilitar el trabajo interdisciplinar, el diseño, formalización y transformación de modelos computacionales y sus posibilidades de apoyo automático a las tareas llevadas a cabo por los actores involucrados en el proceso de diseño y ejecución de artefactos. Estas técnicas han sido utilizadas de forma exitosa en otras áreas, si bien en el caso de la biología sintética tienen sus propias peculiaridades. Como ejemplos concretos de soluciones que pueden aportar las TIC en el campo de la biología sintética se pueden nombrar su capacidad de gestionar la seguridad de la información, debido al estricto control que existe sobre los protocolos y diseños biológicos, el soporte para el diseño colaborativo y multidisciplinar de soluciones o las técnicas de validación y verificación de diseños complejos.

En los últimos años han surgido distintas plataformas que tratan de aplicar algunas de las áreas de la TIC en el desarrollo de protocolos biológicos. Cada plataforma se aproxima al problema con distinta perspectiva. Algunas plataformas buscan ofrecer una interfaz gráfica amigable al biólogo, para que pueda generar protocolos experimentales de forma intuitiva. Otras se centran en la potencia de lenguajes de programación de alto nivel, que permite acceder a múltiples soluciones ya desarrolladas en el ámbito de la informática, pero que quedan lejos del usuario medio de este tipo de sistemas.

La ausencia de una plataforma centralizada, genérica y suficientemente flexible para poder atender las demandas futuras de los biólogos en cuestiones de biología sintética ha sido la principal motivación de desarrollo de este Trabajo Fin de Grado.

Este proyecto está enmarcado dentro del diseño y construcción de un marco de trabajo orientado al desarrollo interdisciplinar de la biología sintética. Por lo tanto, los resultados obtenidos están orientados a una posterior integración con otras funcionalidades y servicios que se están construyendo en la actualidad. En el presente TFG se presenta concretamente el estudio, diseño e implementación parcial de una plataforma de soporte a la biología bajo demanda.

La plataforma propuesta pretende dar soporte al diseño y ejecución de protocolos biológicos, facilitando la colaboración de equipos multidisciplinarios en el diseño y ejecución de

dichos protocolos. La plataforma ofrecerá servicios como transformaciones automáticas de los protocolos y otros específicos de cada laboratorio. Para ello, otro de los principales objetivos del proyecto es proveer a los biólogos y laboratorios de un medio de comunicación y trabajo. Todo esto, además, con características muy importantes para un sistema como la seguridad, la fiabilidad o tolerancia a errores.

Dado que el alcance previamente descrito es muy ambicioso, en este TFG se optimizará la arquitectura funcional de la plataforma para fomentar la flexibilidad y adaptación a distintos perfiles de actores, laboratorios e inclusión de nuevas funcionalidades. Por lo tanto no debe entenderse el resultado tecnológico final como una solución cerrada sino como un marco en el que se permitan integrar nueva soluciones.

3. OBJETIVOS

Objetivo general:

Definición y desarrollo de una plataforma de biología bajo demanda que facilite la gestión y el diseño colaborativo de protocolos biológicos y su envío a los laboratorios adecuados para su correcta ejecución.

Objetivos específicos:

1. Perfiles de diseñadores. La plataforma debe ofrecer a los usuarios la funcionalidad adecuada para personalizar sus perfiles y mostrarlos a otros usuarios a modo de red social.
2. Perfiles de laboratorios. La plataforma debe ofrecer a los laboratorios la funcionalidad adecuada para personalizar sus perfiles, de forma que el sistema pueda ofrecer a los usuarios de forma automática las posibilidades de éxito al enviar un protocolo a un determinado laboratorio.
3. Facilidades de colaboración de usuarios. La plataforma de favorecer la comunicación entre usuarios, proporcionando funcionalidad para la búsqueda condicionada de otros usuarios.
4. Facilidades de actuación sobre protocolos. La plataforma debe atender las necesidades actuales de los biólogos a la hora de desarrollar su trabajo, permitiendo administrar sus diseños y protocolos, interaccionar con otros usuarios a través de ellos y acceder a los servicios que ofrece el sistema y los laboratorios.
5. Gestión de facilidades. La plataforma debe proveer de la funcionalidad adecuada para la creación, modificación y eliminación de facilidades de actuación sobre protocolos.
6. La plataforma debe atender las necesidades de los laboratorios, en cuanto a asociaciones con otros laboratorios a través de federaciones, administración de los servicios ofrecidos y seguridad de los mismos.
7. La plataforma debe tener en cuenta los diferentes aspectos que permitan garantizar la seguridad de usuarios, protocolos y el sistema en general, siempre orientada a la escalabilidad y flexibilidad del mismo.
8. El sistema debe fomentar la interdisciplinariedad de los usuarios, garantizando la seguridad en sus diferentes aspectos.

Alcance

Como se comentó anteriormente, el proyecto debe acomodarse a las características propias de un TFG. Por ello, se ha definido un alcance que permita esta adaptación.

Se priorizarán las siguientes cuestiones:

- Específicamente se van a concretar las principales necesidades de los diferentes tipos de usuarios y la posibilidad de resolverlas mediante las tecnologías y soluciones existentes, integrándolas con la nueva funcionalidad requerida por el sector de la Biología Sintética.
- En el proyecto se cubrirán los aspectos fundamentales del diseño de la plataforma para obtener un sistema flexible y escalable, sobre el que se puedan integrar nuevas funcionalidades de forma asumible por otros desarrolladores, además de los usuarios finales.
- Además de contemplar la seguridad de la información del sistema y permitir probar y validar los diferentes diseños enviados por los biólogos, la plataforma va a ser capaz de realizar un seguimiento en tiempo real de los laboratorios a través de distintos medios.
- Para todo ello, la plataforma busca ser un sistema de almacenamiento de protocolos, con funciones características para los mismos, fomentando el trabajo colaborativo sobre ellos.

Hay que tener en cuenta que una de las ventajas de desarrollar un sistema flexible es que el mismo puede encontrarse en constante evolución. Así, las características de un laboratorio o los tipos de servicios que ofrece, podrán variar con el tiempo y el sistema estará preparado para trabajar con ellos igualmente. Esto va a favorecer la personalización también de los servicios ofrecidos por el sistema (optimización, validación, etc.), cuyos cambios éste podrá asumir eficazmente.

Objetivos operativos:

A partir de los objetivos específicos y el alcance definido, a continuación se definen los objetivos operativos que guiará la definición del plan de trabajo del TFG.

1. Desarrollo de una interfaz gráfica accesible para administradores y los diferentes tipos de usuarios que permita un acceso intuitivo a los diferentes componentes del sistema.

2. Desarrollo de un espacio virtual para biólogos, grupos de investigación, federaciones de investigación, laboratorios, entidades externas de colaboración (minería de datos, publicación, registro, etc.), etc.
3. Almacenamiento de protocolos biológicos, ofreciendo un servicio de almacenamiento de documentos exclusivo para protocolos experimentales, sus resultados asociados, operaciones típicas con esos protocolos, etc.
4. Gestión del ciclo de vida de los diseños y protocolos. La plataforma ofrecerá distintos servicios a sus usuarios: el envío a un laboratorio para la realización del experimento, validación formal del protocolo previo al envío, comparación con otros protocolos gestionados por la plataforma, descarga de protocolos para los que se tenga autorización, etc.
5. Definir la relación con los laboratorios.
 - a. Cada laboratorio tendrá un espacio de laboratorio asociado a su perfil.
 - b. Puede haber federaciones de laboratorios y un laboratorio puede estar en distintas ubicaciones.
 - c. Se debe permitir una opción de búsqueda de laboratorio en función de los requisitos del protocolo experimental y las preferencias del biólogo.
 - d. Información para establecer conexiones seguras con el laboratorio.
 - e. Gestión de credenciales del laboratorio.

4. METODOLOGÍA

El TFG se ha realizado en paralelo con otros TFG que también están integrados en el desarrollo del mismo marco de trabajo. Esto ha favorecido el intercambio de ideas y opiniones entre los diferentes estudiantes que se reparten los diferentes componentes de la plataforma.

Para fomentar el intercambio de ideas y la adaptación de la plataforma a las aportaciones de todos los estudiantes y su director se ha elegido una metodología ágil de desarrollo. El método ágil se ha fundamentado en el prototipado rápido y reuniones semanales con el director y el resto de los compañeros. Esto proporciona beneficios al trabajar en una escala mayor, favoreciendo la interacción con otros ingenieros y la visión de trabajo de un proyecto real. Se desarrollan además habilidades muy útiles para el entorno laboral, algo muy positivo al tratarse de un Trabajo Fin de Grado.

El marco en el que se encuadra este TFG es el de un proyecto de investigación, cuyo grado de incertidumbre a nivel tecnológico es elevado. Por tanto, se ha fomentado la minimización de los riesgos tecnológicos. Para ello, el proyecto de investigación utiliza métodos de programación extrema, orientados a validar nuevos requisitos y soluciones tecnológicas a corto plazo. Posteriormente, se realizan pequeñas integraciones en función de los resultados de validación obtenidos.

El TFG presentado se ajusta a estas exigencias. Aunque los objetivos del proyecto están bien definidos, las decisiones tecnológicas se realizan a muy corto plazo para evitar problemas de disfunción. Además el desarrollo ha sido incremental de forma que se asegura un alineamiento con los requisitos del proyecto de investigación global.

Concretamente, el método de desarrollo adoptado es una combinación de programación extrema y metodología *Scrum*.

La mecánica principal seguida durante todo el desarrollo del TFG consta de los siguientes aspectos:

1. Reunión de planificación, donde se abordan, de forma esquemática, los siguientes temas:
 - Trabajos realizados hasta el momento.
 - Posibilidades de las distintas tecnologías para cumplir con objetivos.
 - Alineación de requisitos con objetivos.
 - Relación del TFG con el objetivo del proyecto de investigación.
 - Relación del TFG con otros TFG afines.
 - Enunciado de objetivos para la semana siguiente.
2. Trabajo semanal en función de los objetivos a corto plazo planteados.
3. Preparación de presentación para la siguiente reunión de planificación.

En general, cada ciclo incluye los tres puntos anteriormente referenciados. En función de las necesidades del TFG en cada momento, los objetivos del punto 1 pueden verse modificados. El ciclo se suele repetir cada semana.

Como se comentó previamente, las reuniones de planificación son, generalmente, conjuntas con otros TFG dirigidos por el mismo director. Aunque los objetivos de cada TFG son distintos, al estar todos alineados con el mismo proyecto de investigación, puede aprovecharse la discusión abierta y constructiva. Se obtienen puntos de vista adicionales que mejoran la obtención de requisitos y las decisiones sobre tecnología.

5. ESTADO DE LA TÉCNICA

5.1 Los protocolos biológicos

El concepto de protocolo biológico utilizado en este TFG se define como el conjunto de entradas, variables de entorno y operaciones que se llevan a cabo en un laboratorio bioquímico para obtener algún resultado esperado. [1]

La formalización de los protocolos biológicos es un reto pendiente de solución en la actualidad. Existen distintas aproximaciones que intentan formalizar estos protocolos. En estas aproximaciones se utilizan las acciones llevadas a cabo por los operarios, reacciones químicas, técnicas propias de biología molecular, especificación de los utensilios empleados, etc.

Para los objetivos de este TFG no es relevante la composición interna de estos protocolos. Sin embargo, sí es necesario conocer el ciclo de vida que tiene un protocolo, además de éste que es el vehículo principal de comunicación entre los distintos agentes del sistema.

En la práctica, aunque funcionalmente existen plataformas avanzadas desde el punto de vista de la biología para gestionar y ejecutar protocolos biológicos bajo los principios de la biología sintética, suponen implementaciones particulares difícilmente integrables con otras soluciones. En este TFG se tendrán en cuenta técnicas de las TIC orientadas a cumplir con los objetivos previamente descritos y que superan las limitaciones de otras plataformas.

5.2 Plataformas Cloud Computing

La computación en la nube (comúnmente conocida como cloud computing) se ha definido como la entrega de servicios de tecnología de información; escalables y elásticos a usuarios finales a través de la red. [2]. La nube abstrae la infraestructura subyacente a la red, y el usuario accede al Servicio que solicita sin percibir su infraestructura de soporte.

Objetivo de la Nube.

Compartir recursos entre consumidores, socios y vendedores de toda la cadena de valor de la nube.

En el caso concreto de este TFG se considera que las ventajas funcionales que proporcionan estas tecnologías pueden ser aplicadas al marco de trabajo de biología sintética expuesto anteriormente. Su potencialidad en escalabilidad, flexibilidad y capacidad de integración pueden ser útiles para conseguir los objetivos planteados. Además estas tecnologías permiten la evolución y adaptación a las necesidades futuras de los usuarios.

Dificultades de implantación del paradigma de la Nube.

Son comunes las sospechas temerosas de aparición de problemas derivados a la pérdida de control de la información por parte de los usuarios. Este recelo también es aplicable al ámbito de la investigación, que es el entorno de aplicación de este TFG. Por ello, es fundamental tomar medidas orientadas a incrementar la seguridad en la Nube, al tiempo que se ofrecen cada vez más y mejores servicios.

Estas cuestiones están siendo investigadas en la actualidad por los sistemas de computación en la nube. Aspectos como la seguridad, la fiabilidad o tolerancia a errores son objeto actual de estudio.

Clasificación e implementación de servicios en la nube

Y por la naturaleza del servicio ofrecido por el proveedor, la nube puede ser: [3]

- Servicio de sólo hardware, entonces se llama **IaaS** (*Infrastructure as-a-Service*) Cloud.
- Si además provee software básico, elástico, flexible y en Grid; entonces la nube es una **PaaS** (*Platform as-a-Service*) Cloud,.
- Y si proporciona ciertas aplicaciones (CRM...) al usuario, entonces es **SaaS** (*Software as-a-Service*) Cloud.

La nube de tipo SaaS será fuente inagotable de progreso para el negocio, la sociedad en general y la investigación en particular. Ello se debe a que los actuales Sistemas de Información se diseñan para tratar una ingente cantidad de datos.. Los datos se gestionan por sistemas heterogéneos, que han madurado desde diferentes disciplinas tecnológicas: motores de búsqueda de información, bases de datos, bases de datos espaciales y geográficos. Toda esta información es hoy una fuente imprescindible de recursos para las organizaciones, la empresa y la sociedad.

En los sistemas basados en la nube los servicios son entidades software que tienden a tener un grado de autonomía elevado y que además están débilmente acoplados con la infraestructura sobre la que se apoyan. Esto facilita su integración con infraestructuras TI existentes, rebasando las clásicas fronteras de los Sistemas de Información y Bases de Datos habituales. Para ello, el Servicio necesita ser: descrito, descubierto, compuesto, y monitorizado.

Una categoría especial de servicio, tanto en computación en la nube como en enfoques más clásicos, es el Servicio Web (SW). Esta forma de diseñar e implementar servicios está revolucionando la comunicación de dominios heterogéneos cuya colaboración eficiente entre diferentes partes es fundamental (como en *e-commerce* y *e-business*). Además están muy integrados en la infraestructura web existente facilitando su despliegue en los entornos actuales. Los SW plantean nuevos retos a la Web, como el descubrimiento automático. Y, a su vez, la Web se enriquece con ellos porque la hace transitar desde una Web de documentos hacia otra nueva Web de documentos y servicios.

Desde el punto de vista del usuario se puede observar cómo la derivación de la lógica de la aplicación a la nube ha llegado a muchos tipos de aplicaciones pertenecientes a varios ámbitos: educación, ocio, salud, justicia, etc. El paradigma del despliegue de servicios en la nube facilita el proceso de despliegue de nuevas versiones así como la integración de aplicaciones en varias plataformas distintas (escritorio, móvil, etc.).

5.3 Plataformas de gestión de protocolos biológicos

Se exponen a continuación algunas plataformas de gestión de protocolos biológicos que ya se encuentran en el mercado.

5.3.1 Emerald

Esta plataforma, desarrollada por el *Emerald Cloud Laboratory* (ECL), busca dar soporte a todos los experimentos estándar disponibles en las ciencias de la vida (*Life sciences*). El ECL también incorpora nuevos experimentos y pruebas personalizadas a través de su programa de *Partnering*. El ECL cobra cada experimento en base al número de pruebas. Los precios varían con el tipo de experimento y los parámetros del estudio.

The screenshot displays the Emerald user interface for configuring a Western Blot experiment. The interface is divided into several sections:

- Left Sidebar:** Contains navigation links for Projects, Samples, Models, Reports, and Data.
- Commands:** A list of actions including Plot Data, Analyze Data, Search Data, Share Data, Publish Data, Compile Experiment (selected), Run Simulation, and Ship Materials.
- Experiments:** A list of experiment types including Transfection, UV/Vis Kinetics, UV/Vis Spectroscopy, UV/Vis Thermodynamics, Vacuum Filtration, Viral Prep, Volume Check, and Western Blot (selected).
- Protein Samples:** A section with a 'Look up sample' button.
- Antibody Samples:** A section with a 'Look up sample' button.
- Variables:** A grid of input fields for various parameters:

Variable	Value
Replicates	1
Denature	False
Sample Volume	10 (μL)
Antibody Volume	5 (μL)
Luminal Volume	150 (μL)
Peroxide Volume	150 (μL)
HRP Volume	10 (μL)
Primary Washes	2
Primary Wash Volume	150 (μL)
Primary Wash Time	30 (m)
Secondary Washes	1
Secondary Wash Volume	150 (μL)
Secondary Wash Time	30 (m)
Separation Time	25 (m)
Separation Voltage	350 (V)
Stacking Time	3 (m)
Stacking Voltage	200 (V)
Exposure Time	100 (s)
Number of Blockings	1
Block Volume	200 (μL)
Block Time	5 (m)
- Execute Experiment:** A button at the bottom right of the variables section.
- Notebook:** A section on the right titled 'Notebook: HIV Expression.nb' containing a code editor with the following code:


```
In[1] = compile[
  experiment[WesternBlot],
```

Figura1. Plataforma de usuario de Emerald

Los experimentos se diseñan a través de su plataforma de usuario, se llevan a cabo de forma remota en su laboratorio y se generan los resultados, que pueden analizarse con posterioridad

5.3.2 Antha

Antha es un lenguaje de programación de alto nivel para biología; diseñado para construir *workflows* (flujos de trabajo) reproducibles y escalables, asociando elementos inteligentes y reutilizables. Estos elementos pueden ser desde un gen o un factor de transcripción hasta una cepa de levadura o la bacteria empleada como anfitrión. Los experimentos pueden generarse asociando *workflows*, potentemente diseñados, planificados, ejecutados y procesados en respuesta a lo **que** el investigador quiere descubrir, pueden dejar el **cómo** de forma abstracta para el usuario si se desea.

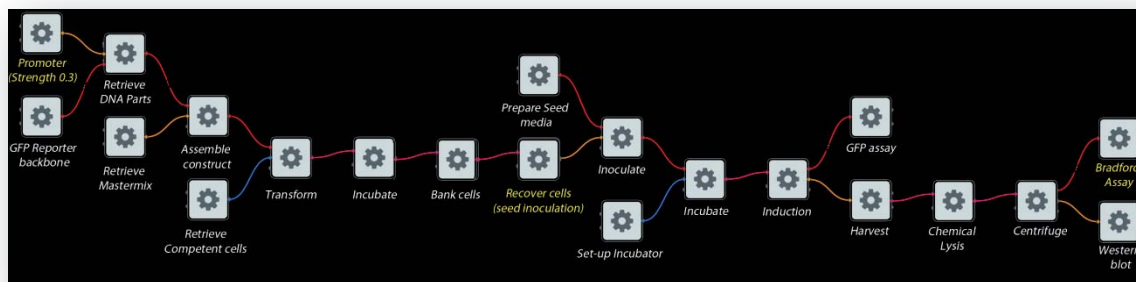


Figura 2. Ejemplo de workflow en Antha

5.3.3 Arcturus.io

Arcturus.io es una comunidad abierta que busca la colaboración entre sus miembros para poder desarrollar los proyectos. Estos se definen seleccionando las condiciones del experimento desde su plataforma online, permitiendo seguir la evolución de los resultados y compartirlos con otros usuarios.

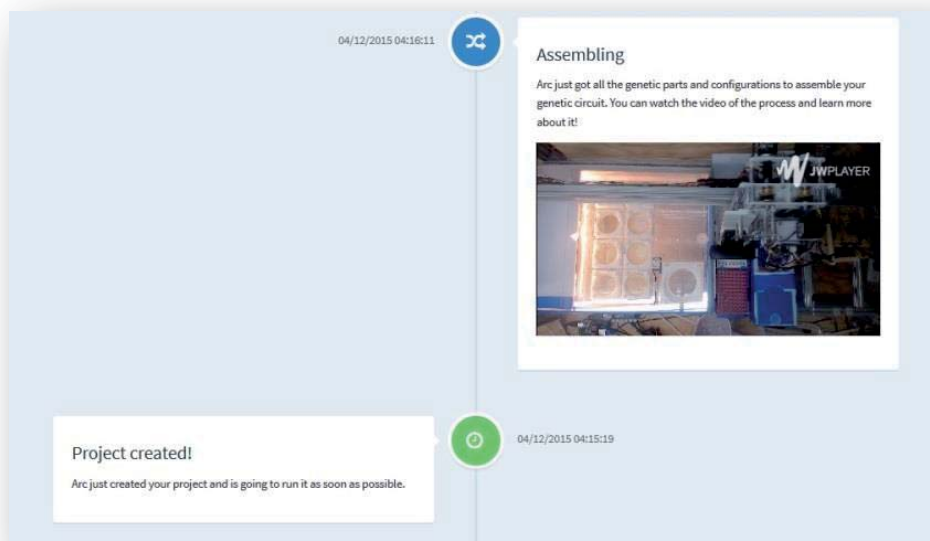


Figura 3. Arcturus.io con vídeo del proceso

5.4 Abstracción del modelo presentado por las plataformas

La plataforma de biología bajo demanda que se va a desarrollar va a integrar algunos de los factores que ya cumplen estas otras plataformas. Aspectos éstos como la interfaz gráfica, que permite generar los experimentos de forma intuitiva, la selección de ciclos de vida adecuados para cada protocolo experimental o la posibilidad de compartir los mismos y trabajar de forma asociada con ellos.

Así, la idea es proporcionar un entorno en el que los usuarios puedan almacenar y gestionar sus protocolos experimentales. Estos protocolos, a su vez, podrán gestionarse de forma intuitiva para operar sobre ellos mediante servicios almacenados en el sistema (optimización, comparación, etc.), acceder a los servicios que pueden ofrecer los diferentes laboratorios o incluso compartirlos con otros usuarios.

Los usuarios además podrán clasificarse según su papel en el sistema, como colaboradores, biólogos, laboratorios, federaciones de laboratorios, etc.; cada uno de ellos con sus operaciones autorizadas pertinentes. Aunque inicialmente se parta con un número limitado de usuarios, el modelo va a permitir una fácil adaptación del mismo a nuevos tipos de usuarios.

El aspecto más importante de esta plataforma es la flexibilidad. A la hora de administrar los diferentes servicios ofrecidos por los laboratorios o almacenados en el sistema, la plataforma ofrece un marco capaz de modificar las características de dichos servicios. Ello permitirá adaptar constantemente el entorno de trabajo a las necesidades futuras, no limitándose únicamente a los aspectos característicos de usuarios o laboratorios, sino generando nuevos flujos de trabajo totalmente diferentes.

Para ello, la plataforma tiene en la escalabilidad una de sus principales metas. La idea general es que sirva de base para futuras ampliaciones sucesivas que vayan surgiendo conforme a la experiencia de usuarios y administradores.

Por supuesto, todo lo anterior debe proporcionarse en un entorno seguro y altamente configurable, que ofrezca suficientes garantías a los usuarios, haciendo de ella un dominio atractivo en cuanto a usabilidad y funcionalidad para el desarrollo de investigaciones y experimentos del ámbito de la biología sintética.

6. PLATAFORMA DE BIOLOGÍA BAJO DEMANDA

6.1 DESCRIPCIÓN

En el desarrollo de esta plataforma de biología bajo demanda, se va a realizar un análisis del sistema, basándose en los objetivos operativos que se han definido.

Así, en el próximo punto, se van a estudiar los diferentes componentes de la plataforma y se presentarán sus requisitos. Concretamente, se busca identificar los requisitos de información y funcionales del sistema.

Para ello, destacamos que el sistema trabaja fundamentalmente con *Usuarios (Biólogos y Colaboradores)*, *Entornos de Ejecución (Laboratorios)*, *Servicios (Servicios de Laboratorios)* y *Documentos (Protocolos)*.

Como prueba del correcto análisis del sistema, se muestra, posteriormente, un caso de uso que ejemplifica la interacción de un usuario con el sistema.

A continuación se expone la arquitectura diseñada para la plataforma de biología bajo demanda, orientada a garantizar la flexibilidad y adaptabilidad del sistema, como se expuso en los objetivos.

En el diseño, se describe y se muestran ejemplos de la interfaz de usuario de la plataforma.

En la implementación se exponen detalladamente las tecnologías empleadas para desarrollar la plataforma, así como las herramientas de desarrollo utilizadas y los aspectos de seguridad que se han considerado.

Por último, se presentan las pruebas que se han definido para mejorar la calidad del sistema implementado.

6.2 ANÁLISIS DEL SISTEMA

Espacio virtual para usuarios

La plataforma de biología bajo demanda tiene entre sus objetivos el almacenamiento de protocolos de diseño, así como de sus servicios. Estos servicios están disponibles para el usuario a través de los laboratorios a los que éste tiene acceso.

Por ello, es necesario proveer al usuario de un *espacio virtual* desde el que poder administrar sus documentos y acceder a la diversa información del sistema.

La primera opción que el usuario tiene disponible es la consulta y modificación de sus datos personales. Los usuarios son agregados al sistema por el administrador. Será posible, posteriormente, introducir funcionalidad al sistema que permita a los usuarios registrarse personalmente en el mismo, pero eso no es objeto de estudio de este trabajo. Lo que sí es posible para un usuario es consultar sus datos personales y modificar parte de ellos, recogidos en un documento en formato JSON asociado a su cuenta.

Dada la flexibilidad del sistema, los usuarios podrían representar multitud de roles diferentes en el mundo real, ya que el modelo así lo permite. En nuestro caso, los usuarios pueden tener el rol de Biólogos o Colaboradores.

La principal herramienta que proporciona el *espacio virtual* a los usuarios es la subida, descarga y actualización de protocolos de diseño biológico. Estos documentos están asociados de forma privada a cada usuario, que puede trabajar con ellos.

Las acciones que un usuario puede efectuar sobre uno de sus documentos dependen de los servicios presentes en la plataforma. Estos servicios pueden ser agregados al sistema por administradores y colaboradores. Representarán acciones varias como compartir el documento, descargarlo, validarlo, optimizarlo, etc.

Por último, los usuarios (biólogos y colaboradores) pueden acceder a través de la sección correspondiente de su *espacio virtual* a los laboratorios presentes en la plataforma y consultar sus datos, su estado actual y los servicios disponibles en los laboratorios a los que están adscritos.

Por otro lado, los usuarios con el rol de colaborador (*colaboradores*) tienen disponibles las mismas características en su *espacio virtual*. Sin embargo, disponen además de capacidad para añadir nuevos servicios para protocolos al sistema, por lo que se les proporciona dicha funcionalidad a través de su espacio.

Almacenamiento de documentos

Los usuarios del sistema pueden usarlo para almacenar, compartir o trabajar con sus documentos. Estos documentos pueden definirse de formas diferentes en función a las necesidades del sistema implementado. En nuestro caso, los documentos representan protocolos de diseño biológico.

Los protocolos de diseño biológico son almacenados en el sistema como documentos llamados simplemente *protocolos*. Estos protocolos son la principal forma de trabajo de los usuarios (biólogos) en la plataforma, ya que contienen la información correspondiente a los diferentes experimentos que desean llevar a cabo. Esta información se almacena en el campo correspondiente del documento en forma de documento con formato JSON.

Los protocolos se almacenan asociados al usuario que los ha subido a la plataforma, aunque se pueden compartir con otros usuarios. Cuando un usuario accede a su *espacio virtual*, dispone de la posibilidad de consultar todos sus protocolos, ya sean propios o compartidos con él.

Los protocolos tienen disponibles una serie de servicios definidos en el sistema, que son accesibles pulsando en cada documento de protocolo, dentro del espacio virtual de un usuario. Ejemplos de estos servicios son: compartir con otros usuarios, descargar, enviar a un laboratorio, validar, optimizar, etc.

Además, cuando un protocolo es procesado, los resultados obtenidos se almacenan asociados al propio documento, pudiendo un protocolo tener asociados los resultados de varios experimentos.

Servicios

Los *servicios* son entidades del sistema que representan las diferentes operaciones que se pueden realizar sobre un documento. Al igual que ocurría anteriormente, el sistema es flexible para ofrecer servicios de múltiples tipos, pero se han concretado en nuestro caso de estudio como *servicios de protocolos*. Estos servicios se ofrecen para interactuar con los protocolos de diseño.

Cuando un usuario ha almacenado un protocolo en el sistema, puede seleccionar llevar a cabo sobre él alguno de los servicios disponibles. Ejemplos de dichos servicios son compartir con otros usuarios, descargar, enviar a un laboratorio, validar, optimizar, etc.

Dado que uno de los objetivos principales de este TfG es desarrollar un sistema flexible y adaptable, los servicios que se ofrecen en la plataforma pueden ser modificados. Así, tanto

colaboradores como administradores podrán cargar nuevos servicios en el sistema y estos últimos pueden además editarlos.

Los *servicios de protocolos* pueden estar asociados a laboratorios si éstos pueden ofrecerlos (por ejemplo, un tipo de experimento que requiere una máquina concreta) o ser de tipo *General* del sistema (enviar a un laboratorio, compartir con otro usuario, etc.).

Finalmente, todos los *servicios* ejecutan una operación que está definida en un campo diseñado para tal fin, almacenando un documento de formato JSON para, nuevamente, dotar de flexibilidad al sistema.

Gestión de entornos de ejecución

Los entornos de ejecución son plataformas que ofrecen servicios a los usuarios para trabajar con sus documentos, al tiempo que se relacionan entre sí. En el caso de nuestra plataforma de biología bajo demanda, dichos entornos de ejecución representan a los laboratorios. Estos laboratorios están representados en el sistema con entidades del mismo nombre y poseen un rol de usuario propio (*Laboratorio*).

Como otros elementos del sistema, están descritos no sólo por sus atributos característicos, sino que existe la posibilidad de asociarles un documento en formato JSON que incremente su funcionalidad.

Los administradores son los encargados de registrar nuevos laboratorios en la plataforma y administrar los existentes. No obstante, los usuarios con el rol de *Laboratorio* pueden agregar nuevos servicios al sistema, que aparecerán asociados al propio laboratorio. De esta forma, los biólogos pueden conocer los servicios que ofrecen los laboratorios a los que están adscritos.

Para que los diferentes biólogos puedan acceder a los servicios de un laboratorio, es necesario que éstos estén adscritos al mismo primero. Por ello, los laboratorios pueden consultar los usuarios existentes en el sistema e inscribirlos en el laboratorio, para autorizarles a usar sus servicios.

Además, los laboratorios pueden asociarse en *Federaciones de Laboratorios*. Las federaciones de laboratorios son grupos asociados de laboratorios que comparten servicios a los cuales pueden acceder todos los usuarios adscritos a cualquiera de los laboratorios miembros de la federación.

6.2.1 Requisitos del Sistema

Catálogo de Requisitos del Sistema – Espacio virtual para biólogos

Formalmente, se definen los siguientes requisitos asociados al *Espacio virtual para biólogos*.

Requisitos de Información

IRQ-0001	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá almacenar la información correspondiente a los <i>usuarios</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre • Descripción • Laboratorios • Datos de usuario • Grupos • Protocolos
Comentarios	Los <i>Datos de usuario</i> son subidos al sistema en un documento en formato JSON.

Requisitos Funcionales

FRQ-0001	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir a usuarios y colaboradores consultar y modificar sus datos</i> .
Comentarios	-

FRQ-0002	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>mostrar una lista con todos sus protocolos de diseño biológico a usuarios y colaboradores.</i>
Comentarios	-

FRQ-0003	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir a los colaboradores subir nuevos servicios de protocolos de diseño biológico al sistema.</i>
Comentarios	-

FRQ-0004	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>mostrar los servicios asociados a un protocolo de diseño biológico.</i>
Comentarios	-

FRQ-0005	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>mostrar a usuarios y colaboradores la lista de laboratorios incluidos en el sistema.</i>
Comentarios	Sólo se mostrarán aquellos laboratorios a los que está autorizado acceder el usuario o colaborador.

FRQ-0006	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir a usuarios y colaboradores unirse a laboratorios o federaciones de laboratorios.</i>
Comentarios	-

Catálogo de Requisitos del Sistema – Almacenamiento de protocolos

Formalmente, se definen los siguientes requisitos asociados al *Almacenamiento de protocolos*.

Requisitos de Información

IRQ-0002	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá almacenar la información correspondiente a <i>los protocolos de diseño biológico</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre • Descripción • Autor • Usuarios • Servicios • Experimento • Resultados
Comentarios	<i>Experimento</i> y <i>Resultados</i> son almacenados en un documento en formato JSON.

Requisitos Funcionales

FRQ-0007	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir almacenar nuevos protocolos de diseño biológico</i> .
Comentarios	Estos protocolos quedan asociados al usuario que los creó.

FRQ-0008	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir compartir un protocolo con otro usuario</i> .
Comentarios	Únicamente su autor o usuarios con acceso compartido podrán hacerlo.

FRQ-0009	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir editar protocolos de diseño biológico existentes.</i>
Comentarios	Únicamente su autor o usuarios con acceso compartido podrán hacerlo.

FRQ-0010	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir seleccionar un servicio adecuado para aplicar sobre el protocolo.</i>
Comentarios	Únicamente su autor o usuarios con acceso compartido podrán hacerlo.

FRQ-0011	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>almacenar los resultados de los experimentos llevados a cabo sobre un protocolo en el campo "Resultados" del mismo.</i>
Comentarios	

Catálogo de Requisitos del Sistema – Servicios de protocolos

Formalmente, se definen los siguientes requisitos asociados a los *Servicios de protocolos*.

Requisitos de Información

IRQ-0003	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá almacenar la información correspondiente a los <i>servicios para protocolos de diseño biológico</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre • Descripción • Autor • Datos • Laboratorio
Comentarios	Los <i>Datos</i> son subidos al sistema por administradores o colaboradores en un documento en formato JSON.

Requisitos Funcionales

FRQ-0012	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir subir nuevos servicios de protocolos</i> .
Comentarios	Únicamente pueden hacerlo administradores, colaboradores y laboratorios.

FRQ-0013	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir editar los servicios de protocolos</i> .
Comentarios	Únicamente pueden hacerlo los administradores o colaboradores y laboratorios, si son autores del mismo.

FRQ-0014	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir asociar servicios de protocolos a laboratorios</i> .
Comentarios	Únicamente pueden hacerlo los administradores o laboratorios.

Catálogo de Requisitos del Sistema – Gestión de laboratorios

Formalmente, se definen los siguientes requisitos asociados a la *Gestión de laboratorios*.

Requisitos de Información

IRQ-0004	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá almacenar la información correspondiente a <i>los laboratorios</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre • Descripción • Ubicación • Contenido • Servicios • Biólogos • Federaciones
Comentarios	

IRQ-0005	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá almacenar la información correspondiente a <i>las federaciones de laboratorios</i> . En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre • Descripción • Miembros
Comentarios	

Requisitos Funcionales

FRQ-0015	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>almacenar la información correspondiente a los laboratorios</i> .
Comentarios	Únicamente los administradores pueden añadir nuevos laboratorios.

FRQ-0016	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir editar la información correspondiente a los laboratorios</i> .
Comentarios	Únicamente los administradores y los laboratorios pueden hacerlo. Éstos últimos exclusivamente su propia información.

FRQ-0017	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir crear nuevas federaciones de laboratorios</i> .
Comentarios	Únicamente los administradores y los laboratorios pueden hacerlo.

FRQ-0018	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir editar federaciones de laboratorios</i> .
Comentarios	Únicamente los administradores y los laboratorios adscritos a ellas pueden hacerlo.

FRQ-0019	Tipos de documentos
Versión	1.0 (13/05/15)
Descripción	El sistema deberá <i>permitir añadir usuarios aun laboratorio</i> .
Comentarios	Únicamente los laboratorios pueden hacerlo.

6.2.2 Interfaces comunicación de plataformas

La *Plataforma de Biología bajo Demanda* es la capa central o *middleware* del proyecto. Las diferentes capas ofrecen sus servicios entre ellas para poder establecer comunicaciones. Emplean para ello el protocolo REST. [4]

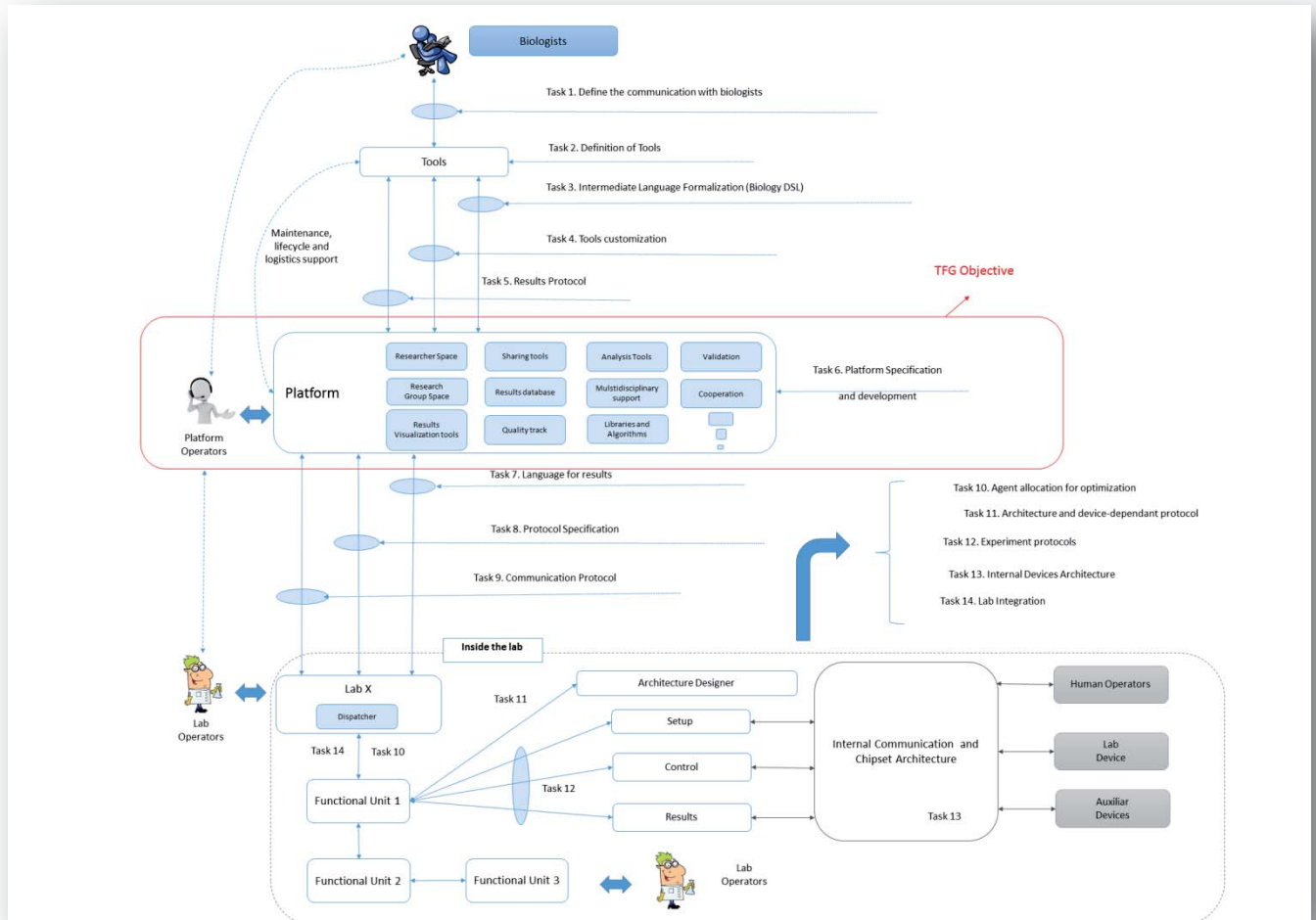


Figura 4. Esquema global de Evoprog [Elaborado por el grupo de investigación LIA – UPM]

Así, las herramientas que emplean los biólogos para diseñar los protocolos biológicos emplean los servicios que ofrece la plataforma. Ésta, a su vez, debe conocer, mediante servicios de la capa inferior, el estado de los laboratorios y sus dispositivos.

6.2.3 Caso de Uso de ejemplo

Para mostrar que se ha realizado el análisis del sistema de forma acertada y su implementación correctamente, se ha diseñado un caso de uso de ejemplo.

El siguiente caso de uso muestra la interacción de un usuario con la plataforma cuando quiere almacenar un nuevo protocolo de diseño e interactuar con él para ejecutar un servicio de un laboratorio sobre el mismo.

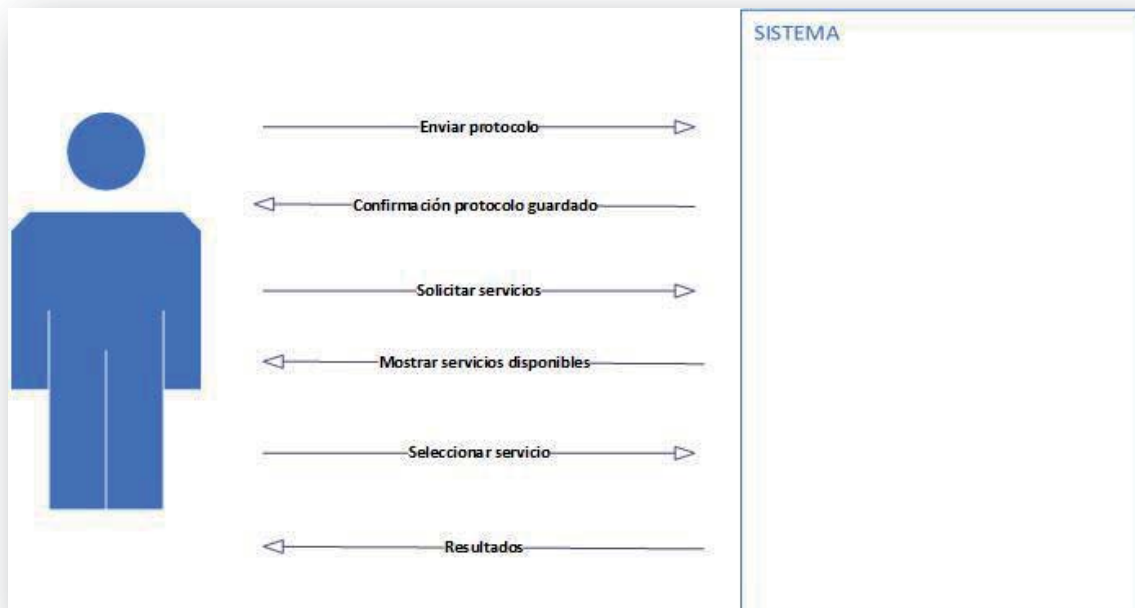


Figura 5. Ejemplo de caso de uso

El usuario solicita enviar un nuevo protocolo al sistema. Posteriormente el sistema confirma al usuario si se ha subido correctamente y muestra el nuevo protocolo entre su lista de protocolos. A continuación, el usuario pulsa sobre el nuevo protocolo y el sistema le muestra los servicios disponibles para el mismo. El usuario escoge qué servicio desea ejecutar sobre el protocolo y el sistema lo lleva a cabo y devuelve los resultados al documento.

6.2.4 Arquitectura del sistema

Dado que uno de los objetivos de este proyecto es obtener un sistema flexible y escalable, se ha decidido diseñar la arquitectura del sistema con este objetivo.

Para ello se ha optado por dividir la arquitectura en tres capas: presentación, negocio y datos. Este modelo de arquitectura cliente-servidor es ampliamente conocido y la tecnología empleada (JavaEE – Glassfish) está orientada a ello.



Figura 6. Programación en capas

Pero, además, surge en el proyecto la necesidad de incrementar aún más esa flexibilidad, permitiendo que la plataforma pueda emplearse en otros ámbitos (por ejemplo, un sistema de intercambio de diseños de protocolos para alumnos del departamento de Inteligencia Artificial).

Por ello, la capa de negocio se ha subdividido en tres capas a su vez: plataforma abstracta, plataforma concreta y productos.

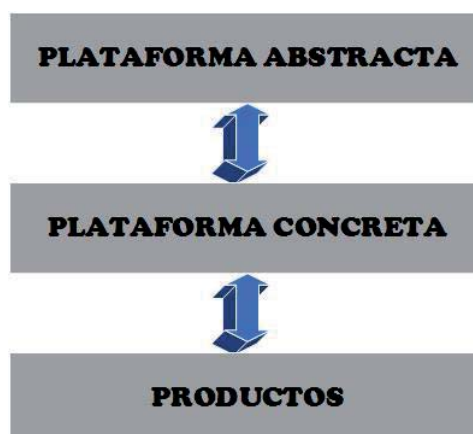


Figura 7. Capa de negocio

Este modelo está basado en un patrón de diseño ampliamente probado: la Factoría Abstracta y se explica a continuación.

6.2.5 Patrón de diseño

Se pretende diseñar el sistema de forma que el acceso al mismo se produzca por un único punto. De esta forma, las capas superiores encuentran un acceso homogéneo a la plataforma, aunque ésta esté dedicada a fines diferentes.

Por ello, surge la necesidad de organizar el sistema como plataformas independientes, cada una de ellas con su propia familia de productos, que dependen de una única plataforma central que controla el acceso.

Para cubrir todas estas necesidades, se ha elegido un diseño basado en el patrón Factoría Abstracta. [5]

Aquí puede verse un diagrama de su diseño:

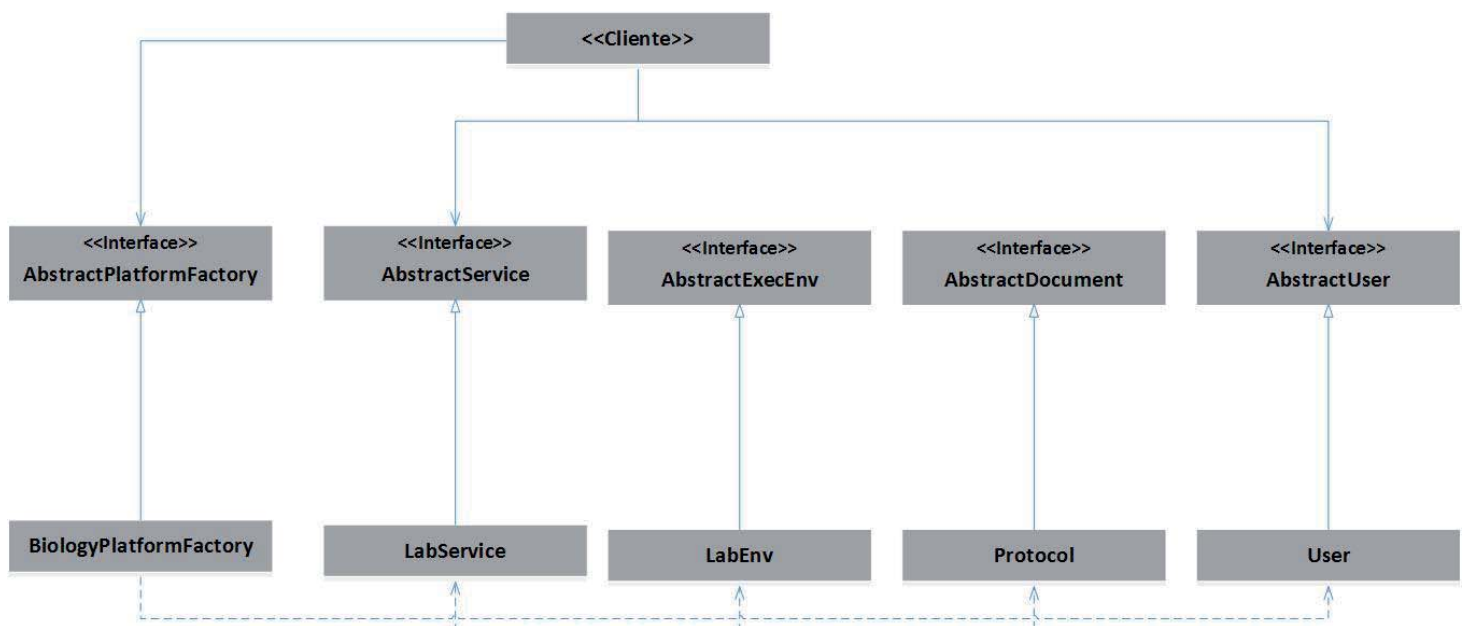


Figura 8. Diseño del sistema

La interfaz *AbstractPlatformFactory* es el punto de entrada común para todos los clientes y se encarga de definir los métodos para crear los productos, siendo la clase concreta la que los crea. Es lo que se denomina *Abstract Factory*. Además, los productos que se generan por la clase concreta implementan todos también “interfaces abstractas”, manteniendo la entrada única al

sistema. Se han definido para esta plataforma como: *AbstractDocument*, *AbstractExecEnv*, *AbstractService* y *AbstractUser*.

Según la orientación del sistema, se pueden generar diferentes fábricas concretas de productos. En este caso, se ha implementado la clase *BiologyPlatformFactory*, que se encarga de implementar los métodos de *AbstractPlatformFactory* y generar los productos concretos. Para el caso de una plataforma de biología bajo demanda, se han definido los objetos concretos *Protocol*, *LabEnv*, *LabService* y *User*.

Como consecuencia de este diseño, se pueden valorar los siguientes aspectos:

- Se oculta a los clientes las clases de implementación, ofreciendo un único punto de entrada al sistema.
- Facilita el intercambio de familias de productos, permitiendo emplear la plataforma para diseñar sistemas con usos diferentes.
- Mejora la consistencia de productos, al obligar a usar objetos de una misma familia.

Como aspecto negativo, se podría destacar que es complicado añadir nuevos tipos de productos, dado que implica extender la interfaz de *AbstractPlatformFactory*. Este caso no se ha considerado que influya demasiado de forma negativa en la elección, dado que el objetivo era diseñar un sistema flexible y sencillo de adaptar. A pesar de ello, si fuera necesario, el esfuerzo requerido para llevar a cabo una actualización de este tipo (agregar un producto nuevo) no es mucho, al ser un patrón de diseño conocido y documentado, por lo que futuros desarrolladores no tendrán excesivas complicaciones en llevarlo a cabo.

6.2.6 Otros aspectos

Cada una de las clases concretas contiene un atributo de cadena de bytes que permite almacenar documentos JSON en la base de datos con funcionalidad adicional, incrementando aún más la flexibilidad y adaptabilidad del sistema. Además, se relacionan entre sí para aportar funcionalidad a la lógica de negocio, siguiendo el siguiente esquema:

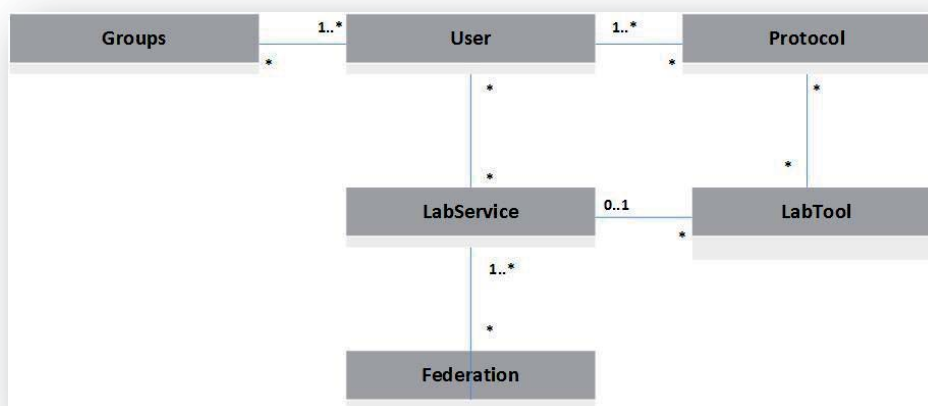


Figura 9. Relación entre entidades

A la hora de implementar la persistencia de las entidades en la base de datos, se ha optado por la tecnología JPA, que incluye JavaEE. Pese a ello, en consonancia con la idea de generar un sistema flexible, se ha diseñado un sistema de persistencia basado en *Data Access Objects* (DAO).

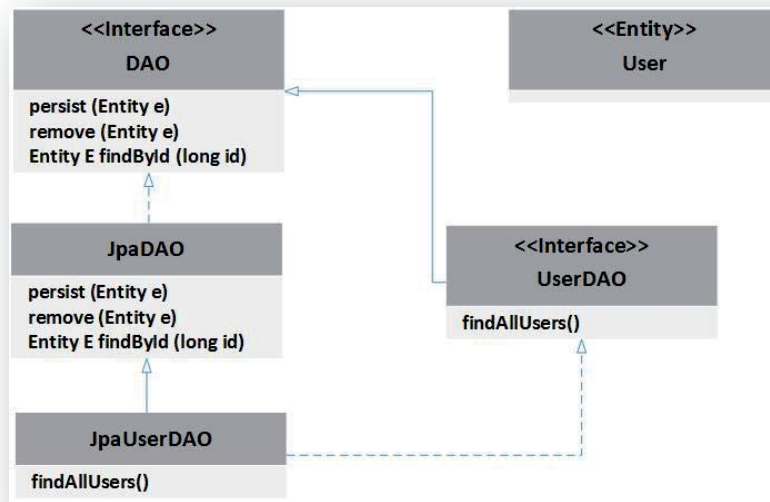


Figura 10. Modelo DAO

Como se puede observar, los objetos heredan de la clase *JpaDAO*, que contiene los métodos comunes a todos los DAO y define el *EntityManager* y *EntityManagerFactory* para manejar las conexiones a emplear. Además, al implementar la interfaz DAO correspondiente a cada objeto, se pueden limitar los métodos que podrá implementar cada DAO.

Los beneficios de este diseño son varios:

- No es necesario escoger el *EntityManager* adecuado en cada caso, ya que está definido previamente.
- Es posible deshabilitar ciertas operaciones para un DAO determinado a través de su interfaz.
- En teoría, al usar la interfaz *DAO* es posible emplear varios sistemas de persistencia, aunque en la práctica se desaconseja por su complejidad.
- Hay un único punto de entrada donde se pueden generar registros o controlar estadísticas de uso.
- Permite centralizar todas las *queries* en un único tipo de entidad, en lugar de tenerlas repartidas por todas las entidades.

Todo ello, por tanto, simplifica el uso del sistema para futuros desarrolladores que pretendan emplear la plataforma para sistemas de características diferentes, al poder emplear un modelo único para todos ellos.

6.2 DISEÑO DEL SISTEMA

6.2.1 Interfaz gráfica para usuarios y administradores

La interfaz gráfica está diseñada para que usuarios y administradores puedan acceder a las funciones y características del sistema de una forma intuitiva y manejable.

El diseño general se asemeja a otros gestores documentales con los que los usuarios seguramente estén familiarizados, lo que favorece a que la curva de aprendizaje sea muy corta. Además, se pretende mostrar una interfaz atractiva al usuario, a la vez que sencilla.

Es importante destacar que, aunque el entorno gráfico es muy similar, los usuarios de diferentes roles tendrán acceso únicamente a aquellas secciones de la plataforma a las que están autorizados.

Así, la interfaz proporciona a los administradores utilidades para:

- Crear/editar usuarios.
- Crear/editar laboratorios.
- Crear/editar servicios.
- Ver protocolos.

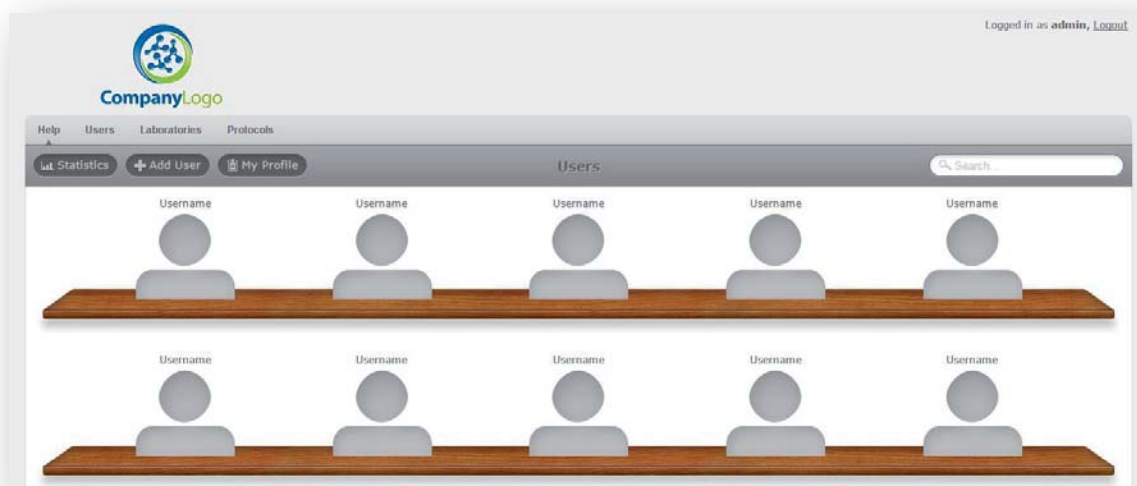


Figura 11. Administración de usuarios

Así, la interfaz proporciona a los biólogos utilidades para:

- Crear/editar protocolos.
- Ver/editar su perfil.
- Ver laboratorios.
- Ver usuarios.
- Ver servicios.

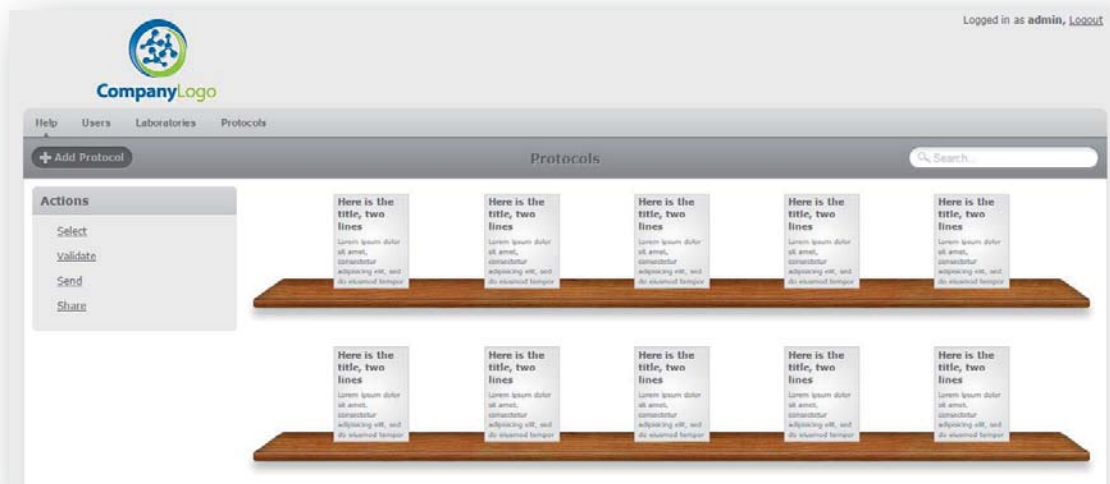


Figura 12. Gestión de protocolos

Así, la interfaz proporciona a los colaboradores utilidades para:

- Crear/editar protocolos.
- Ver/editar su perfil.
- Ver laboratorios.
- Ver usuarios.
- Crear servicios.

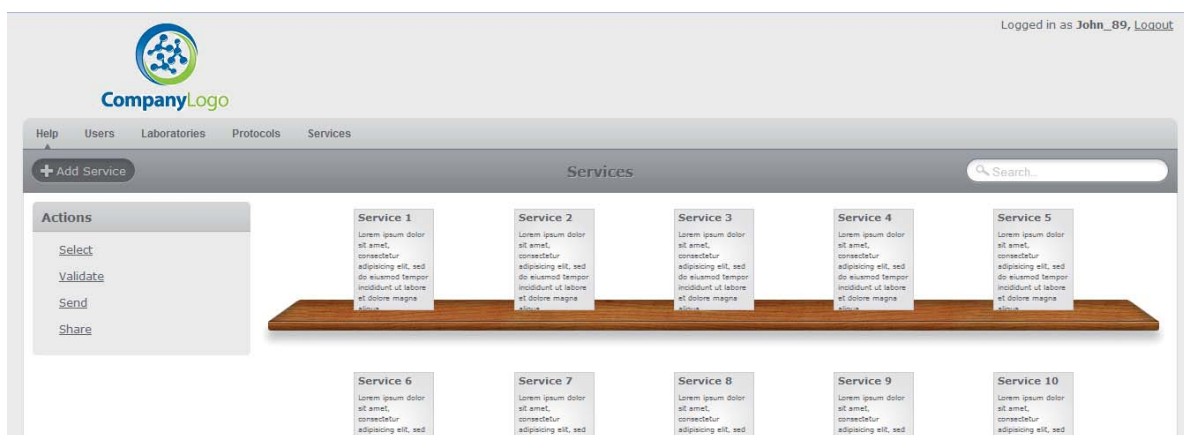


Figura 13. Consulta de servicios

Así, la interfaz proporciona a los laboratorios utilidades para:

- Crear/editar servicios.
- Ver/inscribir usuarios.
- Ver laboratorios.
- Crear federaciones de laboratorios y agregar laboratorios.

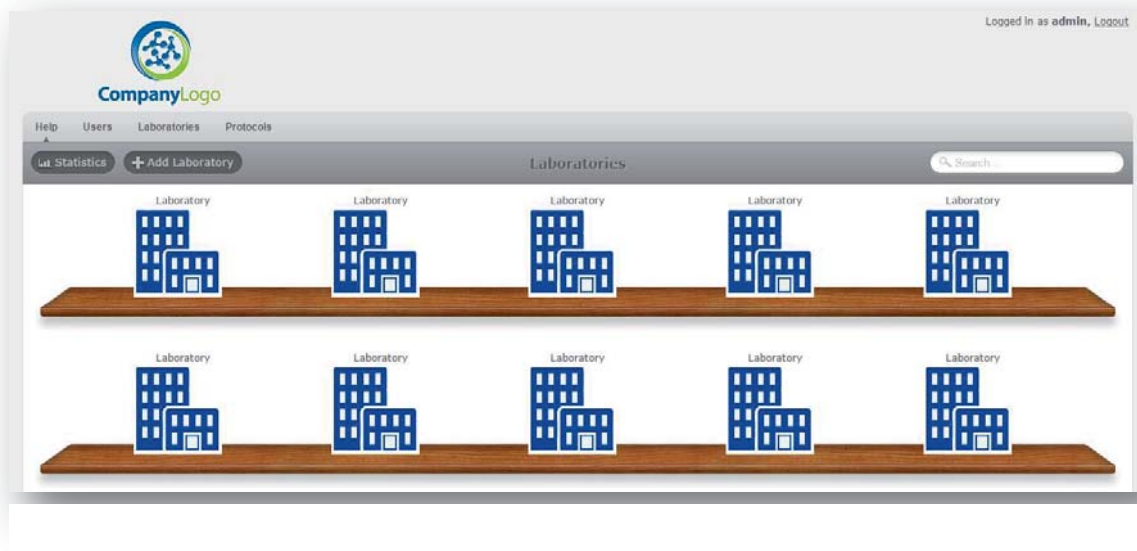


Figura 14. Consulta de laboratorios

6.3 IMPLEMENTACIÓN

Una vez definidas las funciones que debe ser capaz de llevar a cabo una plataforma de biología bajo demanda (Análisis) y que se ha presentado cómo se van a organizar sus diferentes componentes (Diseño), se procede a estudiar la implementación del sistema.

6.3.1 Tecnologías

Es importante señalar que, al tratarse de un proyecto de investigación, una parte importante del mismo ha sido identificar las tecnologías adecuadas para desarrollarlo con el fin de cumplir los objetivos marcados.

Por lo tanto, antes de implementar los diferentes componentes de nuestro sistema, ha sido necesario llevar a cabo un estudio de las diferentes tecnologías que ofrece el mercado. Es en base a esta investigación, como se ha tomado la decisión de trabajar con las tecnologías que se exponen a continuación.

6.3.1.1 JAVA EE/GLASSFISH

En este proyecto surge la necesidad de implementar nuestro sistema de forma que dé soporte a aplicaciones distribuidas, transaccionales y portables. Todo ello, además, con la seguridad y confianza que ofrece la tecnología del lado del servidor. Además, dadas las características inciertas del proyecto en su definición, era primordial que la tecnología escogida fuera lo suficientemente versátil para adaptarse a las posibles soluciones que se implementasen.

Por ello, se optó por una plataforma de desarrollo de aplicaciones empresariales: *Java EE*.

JavaEE de Oracle es una plataforma para aplicaciones empresariales que reúne todas estas características y que, además, es ampliamente conocida en el mundo laboral. Su uso permite cubrir todas las necesidades del proyecto con una tecnología fiable y que ofrece soluciones ya desarrolladas y probadas que mejoran la calidad del resultado final. [6]

Para desplegar el sistema como aplicación web, se ha optado por el servidor *Glassfish* (Release 4.0). Glassfish proporciona un servidor ideal para el despliegue de aplicaciones web desarrolladas para entornos empresariales. [7] Su consola de administración proporciona

múltiples utilidades, entre las que destacan su gestión de la seguridad, algo que se fijó como objetivo prioritario en este proyecto. Además, ofrece soluciones con un alto grado de integración con múltiples sistemas, como puede ser la fuente de datos. Y es que al tratarse éste de un sistema con la flexibilidad como uno de sus principales requisitos, el hecho de que Glassfish sea compatible con bases de datos relacionales, orientadas a documentos o a grafos (entre otras) es un gran punto a su favor.



Figura 15. Consola de administración Glassfish

Por último, destaca la modularidad de Java EE, que permite hacer uso únicamente de aquellos componentes que son necesarios y no incrementar innecesariamente la complejidad de la implementación. Algunos de los principales componentes empleados en este proyecto son *EJB*, *JPA*, *REST*, *JSON* o *JSP*, que se expondrán en puntos posteriores.

6.3.1.2 EJB

Enterprise JavaBeans (EJB) es una tecnología de JavaEE que implementa objetos denominados *Enterprise Beans*. Los Enterprise Beans son objetos escritos en Java que definen componentes del lado del servidor que encapsulan la lógica de negocio de una aplicación. [6]

Los *beans* proporcionan métodos que pueden invocar los clientes para acceder a los servicios que provee la aplicación. Estos beans se ejecutan en el contenedor EJB de Glassfish.

El uso de EJB está especialmente indicado para aplicaciones escalables, con variedad de clientes y cuyas transacciones deben ser seguras. Su uso por tanto, cumple con los objetivos de este proyecto y ha recomendado su elección.

6.3.1.3 JPA

JPA es la API que ofrece JavaEE para implementar la persistencia de sistemas. JPA provee a los desarrolladores de utilidades para el mapeado objeto/relacional a la hora de manejar datos relacionales en aplicaciones Java.

La API está definida en el paquete *javax.persistence*. JPA permite definir las entidades del sistema que se van a persistir en la base de datos. Es decir, permite definir qué objetos se almacenarán en la base de datos de forma coherente y sin errores.

JPA proporciona servicios para diferentes sistemas de gestión de base de datos, ofreciendo compatibilidad con diferentes controladores para llevar a cabo las conexiones con ellos. Estos controladores pueden ser ofrecidos también por el servidor Glassfish, delegando en él la conexión con la base de datos y permitiendo que sea éste el que maneje la seguridad y recursos de la misma.

6.3.1.4 REST

REST es un estilo de arquitectura que se emplea para describir interfaces web simples. Es posible implementar servicios web conforme a la arquitectura REST que permite el acceso a recursos compartidos desde la red.

En este sistema se ha decidido emplear esta arquitectura para definir servicios web de forma simple y clara. Estos servicios web proporcionan comunicación entre las diferentes plataformas que componen el proyecto global y permiten compartir sus recursos a través de una dirección única en la web (**U**niform **R**esource **I**dentifiers).

JavaEE proporciona un módulo para el desarrollo de estos servicios, denominado *JAX-RX*.

6.3.1.5 JSON

JSON es un formato ligero de intercambio de datos, empleado principalmente como alternativa a *XML*, debido a la simplicidad a la hora de definir analizadores lógicos (*parser*) para él.

Además, permite encriptación de su contenido, lo que cumple el objetivo de seguridad de este proyecto. El hecho de poder contar con documentos seguros y con un alto grado de personalización para almacenar la información de las diferentes entidades del sistema, hace que su elección cumple otro de los principales objetivos que se han planteado: la flexibilidad.

JavaEE incluye una API para generar analizadores sintácticos, transformar y consultar datos JSON, confirmando aún más lo acertado de su elección como plataforma de trabajo.

6.3.1.6 JSP

JSP (Java Server Pages) es una tecnología que permite crear webs dinámicas basadas en HTML, XML u otros formatos. JSP permite trabajar en Java, lo que supone poder desplegar las aplicaciones en diferentes tipos de plataformas, empleando su máquina virtual. [8]

JSP añade gran funcionalidad a la página web dinámica, permitiendo manejar el acceso a datos de la aplicación web.

Se ha optado por el uso de esta tecnología dada su funcionalidad y sencillez y, aunque JavaEE ofrece alternativas más potentes como *JSF*, dado que el objetivo de este proyecto no era el diseño de una interfaz potente, JSP ofrece todo lo necesario de una forma más simple.

Aun así, el uso de JSF en el futuro es una opción plausible, con no demasiadas modificaciones sobre el código ya implementado.

6.3.2 Herramientas de desarrollo

6.3.2.1 Eclipse

El entorno de desarrollo que se ha decidido emplear para este proyecto es *Eclipse Luna Service Release 2 (4.4.2)*. Esta versión también se conoce como *Eclipse Java EE IDE for Web Developers*.

La plataforma Eclipse proporciona un entorno adecuado para el desarrollo de aplicaciones empresariales JavaEE y su uso es muy extendido.

Al ser de código abierto, tiene una comunidad de usuarios muy activa, lo que favorece su mejora continua y encontrar soluciones a los diferentes problemas que pueden surgir durante el desarrollo de un proyecto.

Por todo ello, se ha escogido como plataforma de desarrollo para este TFG.

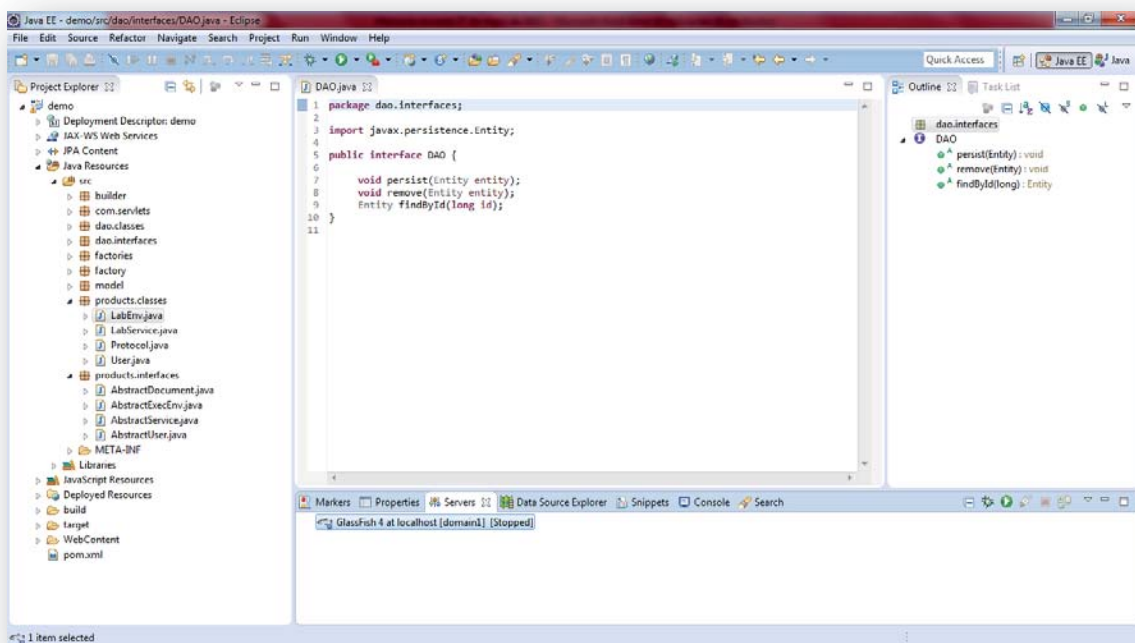


Figura 16. Plataforma Eclipse

6.3.3 Seguridad

6.4.3.1 Mecanismos de Seguridad en Java EE/ Glassfish

Este sistema, como cualquier sistema que ofrece recursos a usuarios a través de la red, necesita ser protegido. A ello hay que sumar que se opera con datos que pueden estar sujetos a las diferentes leyes de protección de datos, puesto que se trata de información referente a investigaciones privadas o datos confidenciales.

JavaEE despliega las aplicaciones web dentro de contenedores. Estos contenedores están provistos de dos tipos de seguridad: declarativa y programática. [9]

La seguridad declarativa define los requisitos de seguridad de un componente de la aplicación mediante descriptores de despliegue o anotaciones. Estas anotaciones se añaden al código y son interpretadas por Glassfish para gestionar la seguridad.

Los descriptores de despliegue son consultados por JavaEE en el momento de ejecución y permiten definir restricciones de seguridad sin tener que modificar el código, únicamente accediendo al fichero de configuración *wem.xml* (en el caso de aplicaciones web) y *ejb-jar.xml* para aplicaciones que emplean Enterprise Beans.

Glassfish, por otro lado, proporciona diversos mecanismos de seguridad para la autenticación y autorización de usuarios, gestión de permisos, control de seguridad a nivel de transporte y de mensajes, etc. Todos estos mecanismos son configurables a través de la consola de administración de Glassfish y se coordinan con la aplicación JavaEE a través de los diferentes ficheros de configuración, como *glassfish-web.xml*. [10]

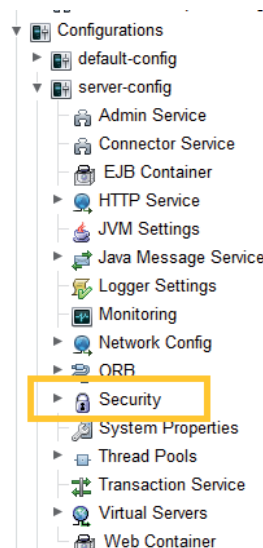


Figura 17. Administración de seguridad Glassfish

Entre los distintos métodos de seguridad que proporciona Glassfish, se encuentran los certificados. Los certificados son documentos electrónicos cifrados que permiten establecer una conexión segura entre el cliente y el servidor. En este proyecto se emplea la identificación de usuarios a través de certificados digitales que trabajan sobre una capa de conexión segura *SSL*.

El intercambio de datos entre cliente y servidor se pueden llevar a cabo a través de mensajes, cuya seguridad también se puede administrar con Glassfish y es posible emplearla, por ejemplo, para la gestión de protocolos biológicos por parte de los usuarios.

Por otro lado, las características propias de la autenticación y autorización de usuarios propias de JavaEE y Glassfish nos permiten administrar los usuarios de forma organizada, a través de *Groups* y *Roles*.

Los *Groups* clasifican los usuarios en grupos, lo que permite limitar el acceso del mismo a las diferentes partes del sistema. Por ejemplo, en este proyecto existen grupos de usuarios, colaboradores, administradores y laboratorios.

Los *Roles* definen los permisos autorizados para cada grupo de usuarios. Así, es posible definir que tanto Usuarios como Colaboradores puedan acceder a los servicios de los laboratorios, pero únicamente los Colaboradores podrán acceder recurso correspondiente para cargar nuevos servicios al sistema.

Estas cualidades de Glassfish, permiten extraer toda la complejidad del manejo de usuarios de la lógica de la aplicación y administrarla desde el servidor a través de operaciones más sencillas e intuitivas.

6.4 PRUEBAS

Respetando la metodología seguida, las pruebas del sistema se han ido realizando sobre las diferentes versiones entregadas.

Así, en los primeros pasos de investigación, las pruebas se han centrado en pruebas de conceptos sobre la tecnología, como, por ejemplo:

- Prueba de concepto de instalación Glassfish, despliegue de una aplicación “Hola Mundo”.
- Prueba de concepto de seguridad Glassfish, ejemplo de seguridad de aplicación y mensajes.
- Prueba de concepto de seguridad Glassfish, ejemplo de usuarios, grupos y reinos.
- Prueba de concepto JavaEE, ejemplo de persistencia JPA.
- Prueba de concepto JavaEE, carga de documentos en el sistema.
- Prueba de concepto JavaEE, seguridad con certificados.

Una vez tratados los diferentes aspectos de la tecnología, se procedió a implementar el proyecto en sí. A partir de ese momento, se fueron desarrollando versiones completas, incrementando la funcionalidad de las mismas. Durante todo este proceso, se procedió a la verificación de las diferentes versiones, comprobando si cumplían los requisitos exigidos.

La validación de las diferentes versiones se realizó a través de pruebas de usuario, entre los diferentes miembros del grupo de trabajo.

Por último, es preciso señalar que, puesto que se trata de un proyecto de investigación, los requisitos han sufrido numerosas modificaciones durante el proceso, debiendo proceder a un nuevo análisis de los mismos y su consecuente desarrollo.

7. CONCLUSIONES

Tras el trabajo llevado a cabo para la realización de este Trabajo Fin de Grado, se puede concluir que tanto las tecnologías empleadas como la metodología seguida han sido acertadas, dado que ha sido posible cubrir los objetivos previamente planeados en el plazo programado. También queda demostrada la utilidad de metodologías hábiles para un acercamiento preciso a la solución buscada minimizando los riesgos tecnológicos propios de entornos con alta incertidumbre.

Para terminar, cabe destacar que el modelo final de la plataforma no quedará completo con este Trabajo Fin de Grado. Ni siquiera con la combinación de los trabajos de los diferentes estudiantes trabajando en ella. Y eso será algo positivo, pues indicará que se ha aportado una base lo suficientemente flexible como para permitir el trabajo futuro de otros desarrolladores, administradores o estudiantes.

Incluso, bastaría con modificar algunos de los aspectos personalizables del sistema para ofrecer a usuarios de otras disciplinas totalmente diferentes a la Biología Sintética una plataforma que les aporte los beneficios que puede aportar a ésta.

Ello demuestra que los objetivos de diseñar una plataforma adaptable a múltiples disciplinas y sujeta a cambios en el futuro se han cumplido.

8. BIBLIOGRAFÍA

- [1] Sage Publications, «Protocols for Experiments,» [En línea]. Available: http://www.sagepub.com/upm-data/42770_11.pdf. [Último acceso: 20 Abril 2015].
- [2] Fonebell, «Fonebell,» [En línea]. Available: <http://www.fonebell.in/cloud-computing-technology-new-revolution/>. [Último acceso: 15 abril 2015].
- [3] C. C. Rodríguez, «<http://sinbad.dit.upm.es>,» Junio 2010. [En línea]. Available: <http://www.upm.es/sfs/Rectorado/Gabinete%20del%20Rector/Notas%20de%20Prensa/2010/2010-06/documentos/CloudComputing.pdf>. [Último acceso: 22 Abril 2015].
- [4] The Open University, «E-business technologies: foundations and practice,» 2008.
- [5] E. E. A. GAMMA, PATRONES DE DISEÑO: ELEMENTOS DE SOFTWARE ORIENTADO A OBJETOS REUTILIZABLES, ADDISON-WESLEY, 2002.
- [6] ORACLE, «Java Platform, Enterprise Edition: The Java EE Tutorial,» [En línea]. Available: <https://docs.oracle.com/javaee/7/tutorial/title.htm>.
- [7] Oracle Corporation, «GlassFish Server Open Source Edition. Application Development Guide. Release 4.0».
- [8] K. S. & B. B. BRYAN BASHAM, Head First SERVLETS & JSP, O'Reilly, 2008.
- [9] B. LÖWENSTEIN, JAVA EE SECURITY, ENTWICKLER.PRESS .
- [10] Oracle Corporation, «GlassFish Server Open Source Edition. Security Guide. Release 4.0,» 2013.
- [11] A. Otero, «Tutorial Básico de Java EE,» 2010.

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Wed Jun 10 16:52:03 CEST 2015
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)